CLAIMS:

1-6. (Canceled)

7. (Original): A method for entering at least one command into a plurality of command queues, comprising:

determining which command queue of the plurality of command queues at least corresponds to the at least one command;

entering the at least one command into the command queue that corresponds;

upon entering the at least one command, taking a snapshot of the order of each of the plurality of command queues;

updating a valid bit to indicate that a queue location is valid;

determining if the command is dependent on any other commands to indicate if dependencies exist; and

if any dependencies exist, updating at least one dependency in a dependency bit.

8. (Original): The method of Claim 7, wherein at least one command queue of the plurality of command queues is a strict order queue.

9. (Original): The method of Claim 8, wherein the method further comprises if the command at least corresponds to the strict order queue, entering the at least one command into the strict order queue in a location indicated by a newest entry pointer.

10. (Original): The method of Claim 7, wherein at least one command of the plurality of command queues is a stack down order queue.

11. (Original): The method of Claim 10, wherein the method further comprises:

if the command at least corresponds to the stack down order queue, entering the at least one command into the stack down order queue; and

upon entering, updating an identification bit to at least track the command.

12-14. (Canceled)

15. (Original): A computer program product for entering at least one command into a plurality of command queues, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer program code for determining which command queue of the plurality of command queues at least corresponds to the at least one command;

computer program code for entering the at least one command into the command queue that corresponds;

upon entering the at least one command, computer program code for taking a snapshot of the order of each of the plurality of command queues;

computer program code for updating a valid bit to indicate that a queue location is valid;

computer program code for determining if the command is dependent on any other commands to indicate if dependencies exist; and

if any dependencies exist, computer program code for updating at least one dependency in a dependency bit.

16. (Original): The computer program product of Claim 15, wherein at least one command queue of the plurality of command queues is a strict order queue.

17. (Original): The computer program product of Claim 16, wherein the computer program product further comprises if the command at least corresponds to the strict order queue, computer program code for entering the at least one command into the strict order queue in a location indicated by a newest entry pointer.

18. (Original): The computer program product of Claim 15, wherein at least one command of the plurality of command queues is a stack down order queue.

19. (Original): The computer program product of Claim 18, wherein the method further comprises:

if the command at least corresponds to the stack down order queue, computer program code for entering the at least one command into the stack down order queue; and

upon entering, computer program code for updating an identification bit to at least track the command.

20-30. (Canceled)

31. (New): A method for maintaining order between multiple command queues with different ordering requirements, the method comprising:

receiving a command at a unit pipeline in a multi-queue system, wherein the multi-queue system has a plurality of command queues comprising a strict order queue and a stack down order queue;

directing the command to a corresponding command queue within the plurality of command queues;

entering the command into the corresponding command queue as a command entry, wherein the command entry comprises a command portion, a validation bit portion, and a set of dependency bits;

upon entering the command into the corresponding command queue, taking a snapshot of an opposing command queue within the plurality of command queues; and

setting the validation bit portion and the set of dependency bits in the command entry based on the snapshot of the opposing command queue.

32. (New): The method of claim 31, wherein the corresponding command queue is the strict order queue.

33. (New): The method of claim 32, the method further comprising:

entering the command entry into the strict order queue in a location indicated by a newest entry pointer.

34. (New): The method of claim 31, wherein the corresponding command queue is the stack down order queue.

35. (New): The method of claim 31, further comprising:

determining a next command to execute from the plurality of command queues;

executing the next command;

generating a retire signal; and

updating the validation bit portion and the set of dependency bits of remaining command entries within the plurality of command queues based on the retire signal.

36. (New): The method of claim 35, wherein the retire signal further comprises command identification.

37. (New): The method of claim 36, wherein updating the validation bit portion and the set of dependency bits of a given remaining command entry comprises:

clearing any dependencies on the executed next command in the set of dependency bits in the given remaining command entry.

38. (New): A multi-queue system, comprising:

a plurality of command queues comprising a strict order queue and a stack down order queue;

a unit pipeline for receiving a command;

means for directing the command to a corresponding command queue within the plurality of command queues;

means for entering the command into the corresponding command queue as a command entry, wherein the command entry comprises a command portion, a validation bit portion, and a set of dependency bits;

means for taking a snapshot, upon entering the command into the corresponding command queue, of an opposing command queue within the plurality of command queues; and

means for setting the validation bit portion and the set of dependency bits in the command entry based on the snapshot of the opposing command queue.

39. (New): The multi-queue system of claim 38, wherein the corresponding command queue is the strict order queue.

40. (New): The multi-queue system of claim 39, further comprising:

means for entering the command entry into the strict order queue in a location indicated by a newest entry pointer.

41. (New): The multi-queue system of claim 38, wherein the corresponding command queue is the stack down order queue.

42. (New): The multi-queue system of claim 38, further comprising:

means for determining a next command to execute from the plurality of command queues;

means for executing the next command;

means for generating a retire signal; and

means for updating the validation bit portion and the set of dependency bits of remaining command entries within the plurality of command queues based on the retire signal.

43. (New): The multi-queue system of claim 42, wherein the retire signal further comprises command identification.

44. (New): The multi-queue system of claim 43, wherein the means for updating the validation bit portion and the set of dependency bits of a given remaining command entry comprises means for clearing any dependencies on the executed next command in the set of dependency bits in the given remaining command entry.